

## CLAIMS

What is claimed is:

1. A method for enabling COBOL programs for distributed processing, comprising:  
    providing a technical layer for use by a COBOL program, the technical layer  
        enabling a distributed processing module;  
    providing a COBOL program; and  
    employing, by the COBOL program, the distributed processing module to enable  
        the COBOL program to perform distributed processing, the COBOL  
        program and the technical layer operating in the same runtime  
        environment.
2. The method of Claim 1, wherein the technical layer is further defined as a COBOL technical layer.
3. The method of Claim 1, wherein the technical layer is further defined as a compiler enabling a plurality of distributed processing modules.
4. The method of Claim 1, wherein the technical layer is further defined as a pre-compiler enabling a plurality of distributed processing modules.
5. The method of Claim 1, wherein the method further comprises providing a mainframe computer system wherein the COBOL program and technical layer are provided.
6. The method of Claim 1, wherein the method further comprises:  
    providing a library of callable routines in the technical layer; and  
    linking the library of the technical layer to the COBOL program.

7. The method of Claim 6, wherein providing the library includes:  
generating an object code using the technical layer; and  
linking the object code of the technical layer to the COBOL program.
8. The method of Claim 6, wherein a source code of the technical layer is written in COBOL language.
9. The method of Claim 6, wherein a source code of the technical layer is written in Assembly language.
10. The method of Claim 1, wherein the technical layer is further defined as including a plurality of distributed processing modules.
11. The method of Claim 10, wherein the plurality of distributed processing modules of the technical layer are further include at least a socket module,
12. The method of Claim 11, wherein the COBOL program and the technical layer operate in the runtime environment defined as at least the distributed processing module providing at least some commonly used functions and variables for use by the COBOL program while the COBOL program is operating.
13. The method of Claim 1, wherein providing the COBOL program further comprises:  
writing, in COBOL language, the COBOL program to perform a distributed processing task.
14. The method of Claim 1, wherein the method further comprises performing, by the COBOL program using the distributed processing module of the technical layer, the distributed processing task.

15. The method of Claim 1, wherein the distributed processing module is further defined as a COBOL program subsystem.
16. The method of Claim 1, wherein the distributed processing module is further defined as a COBOL program library routine call.
17. The method of Claim 1, wherein the distributed processing module is further defined as a COBOL program compiler enabled function.

18. A system for enabling distributed and asynchronous processing by COBOL programs on a computer, comprising :
- a computer system;
  - a COBOL extension layer enabling at least one module operable for a distributed and asynchronous processing task; and
  - a program written in COBOL programming language for the computer system, the program employing the module to perform the distributed and asynchronous processing task on the computer system.
19. The system of Claim 18, wherein the COBOL extension layer is further defined as enabled by a compiler.
20. The system of Claim 18, wherein the COBOL extension layer is further defined as a library linked to the COBOL program, the library having at least one routine callable from the COBOL program.
21. The system of Claim 18, wherein the computer system is a mainframe.
22. The system of Claim 18, wherein the distributed and asynchronous processing task is further defined as at least one of a socket task, a shared memory task, a thread task and a task, a signal handler task, an events task, a semaphore task, and a mutex task.

23. A COBOL compiler for enabling COBOL programs to perform distributed and asynchronous processing, the COBOL compiler comprising:  
an engine to compile a COBOL program; and  
a distributed and asynchronous processing module to enable distributed and asynchronous processing by the COBOL program.
24. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for socket communications.
25. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for pipe communications.
26. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for sharing memory between COBOL programs.
27. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for processing threads.
28. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program to use a queue.

29. The COBOL compiler of Claim 28, wherein the queue is further defined as a message queue.
30. The COBOL compiler of Claim 28, wherein the queue is further defined as a memory queue.
31. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for operating system signal handling.
32. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program for managing events.
33. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program to use semaphores.
34. The COBOL compiler of Claim 23, wherein the distributed and asynchronous processing modules of the compiler includes a module to enable the COBOL program to use mutexes.

35. A method for enabling COBOL programs for asynchronous processing, comprising:
- providing a technical layer for use by a COBOL program, the technical layer enabling an asynchronous processing module;
  - providing a COBOL program; and
  - employing, by the COBOL program, the asynchronous processing module to enable the COBOL program to perform asynchronous processing, the COBOL program and the technical layer operating in the same runtime environment.
36. The method of Claim 35, wherein the technical layer is further defined as a COBOL technical layer.
37. The method of Claim 35, wherein the technical layer is further defined as a compiler enabling the asynchronous processing module.
38. The method of Claim 35, wherein the technical layer is further defined as a pre-compiler enabling the asynchronous processing module.
39. The method of Claim 35, wherein the method further comprises providing a mainframe computer system wherein the COBOL program and technical layer operate in the runtime environment.